ISCA Archive
http://www.isca-speech.org/archive

Odyssey 2008: The Speaker and
Language Recognition Workshop
Stellenbosch, South Africa
January 21-24, 2008

Kernel Combination for SVM Speaker Verification

Réda Dehak[1], *Najim Dehak*[2,3], *Patrick Kenny*[2], *Pierre Dumouchel* [2,3]

[1]Laboratoire de Recherche et de Développement de l'EPITA (LRDE), Paris, France
[2]Centre de recherche informatique de Montréal (CRIM), Montréal, Canada
[3]École de Technologie Supérieure (ETS), Montréal, Canada
reda.dehak@lrde.epita.fr, {najim.dehak,patrick.kenny,pierre.dumouchel}@crim.ca

Abstract

We present a new approach to construct kernels used on support vector machines for speaker verification. The idea is to learn new kernels by taking linear combination of many kernels such as the Generalized Linear Discriminant Sequence kernels (GLDS) and Gaussian Mixture Models (GMM) supervector kernels. In this new linear kernel combination, the weights are speaker dependent rather than universal weights on score level fusion and there is no need to extra-data to estimate them. An experiment on the NIST 2006 speaker recognition evaluation dataset (all trials) was done using three different kernel functions (GLDS kernel, Gaussian and linear GMM supervector kernels). We compared our kernel combination to the optimal linear score fusion obtained using logistic regression. The optimal weights was trained on all 1conv4w-1conv4w trials of NIST-SRE 2005. Testing on NIST-SRE 2006 database, we had an equal error rate of $\simeq$ 5.9% using the kernel combination method which is better than the optimal score fusion system ($\simeq$ 6.1%).

1. Introduction

In current speaker verification systems, the best results are obtained by fusing the scores of several subsystems. Many score fusion techniques are proposed: naive Bayes [1], Neural Network (NN) [1], Support Vector Machines (SVM) [2] and, logistic regression [3, 4]. A problem with all of these techniques, except for naive Bayes, is that held-out data is needed to properly weight the contributions of the individual subsystems. It is well known that the performance of the fused system can degrade drastically if there is a mismatch between the held-out data which serves to estimate the fusion weights and the data on which the system is tested. Another weakness of score-level fusion is that it is based on a single set of fusion weights, common to all target speakers. Clearly, it would be desirable to allow the fusion weights to vary from one speaker to another if speaker-dependent fusion weights could be reliably estimated.

Speaker verification systems based on support vector machines lend themselves to another type of "fusion", namely combination at the kernel level, which does not suffer from either of these drawbacks. Given a set of kernels, we can construct a new kernel for each target speaker by taking a linear combination of these kernels. There is no difficulty in principle in making the coefficients in this linear combination speaker-dependent. In fact, the coefficients can be estimated for each target speaker using the same set of impostors as serve to estimate the speaker-dependent hyperplane separator in SVM training. This also dispenses with the need for held-out data to estimate score-level fusion weights.

The paper is organized as follows: Section 2 presents score fusion methods. Section 3 presents the principal aspect of SVM method. We describe the approach of kernel combination method in section 4. The kernel functions used in our experimentation are presented in Section 5 and the application of kernel combination in speaker verification task in Section 6. Section 7 presents our experiments on NIST-SRE 2005 and 2006 databases. We conclude the paper in Section 9.

2. Score Fusion Methods

The objective of score fusion method is to fuse multiple subsystems into a single effective one. By score fusion, we mean that the resulting output score of the fused system is obtained from the combination of scores of the several subsystems. Many approaches have been used to deduce the resulting score. In [5], the authors used a perceptron classifier, the fusion classifier is trained to minimize the DCF. Kajarekar [6] used a linear combination with equal weight of the scores of four different SVM systems.

The most popular approach used during the last NIST Speaker Recognition Evaluation (SRE)[1] campaign was the linear score fusion with a logistic regression training method [4]. The resulting score of linear score fusion is computed as:

$$s_f(x) = w_0 + \sum_{l=1}^{M} w_l s_l(x) \tag{1}$$

where $s_l(x)$ is the l^{th} subsystem score for test x, M is the number of subsystems which are fused, $w = (w_0, w_1, ..., w_M)^t$ a real vector of weights and $s_f(x)$ is the fused output score.

The optimal weights vector is obtained by logistic regression training on a fusion dataset. The goal of logistic regression is to find the optimal weights vector such as the performance of the fused system will be better than the subsystems performance. The fusion dataset should not be used during the development of the subsystems. If, for example, the training dataset had been used to train NAP/eigenchannel, then it will not be suitable for training the fusion weights because the scores produced by these subsystems would be over-optimistic [4]. As a consequence, we need much more training data to develop a fused system.

3. Support Vector Machines

An SVM [7] is a two-class classifier based on a hyperplane separators. It works by embedding the data into a Hilbert space (feature space), and searching for a linear separator in this

[1] http://www.nist.gov/speech/tests/spk/2006/

space. Usually, the feature space $\mathcal{F}$ has high dimensionality (potentially infinite), and is non linearly related with a mapping function ϕ to the original input space $\mathcal{X}$. The mapping is performed implicitly, by specifying the inner product between each pair of points (x_1, x_2) rather than giving their corresponding coordinates $\phi(x_1), \phi(x_2)$ in the feature space. Given an observation $x \in \mathcal{X}$ and a mapping function ϕ, an SVM discriminant function is given by:

$$f(x) = \langle w \ , \ \phi(x) \rangle + \ b \tag{2}$$

where $\langle w \ , \ \phi(x) \rangle$ represents the scalar product of the two vectors w and $\phi(x)$. (w, b) are the linear separator parameters.

Exploiting the kernel function $k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$ and the fact that the weight vector w can be expressed as a linear combination of a subset of training points ($\{(x_i, y_i) \mid i = 1...M\}$ called support vectors):

$$w = \sum_{i=1}^{M} y_i \alpha_i \phi(x_i) \tag{3}$$

where $y_i = \pm 1$ and α_i represent respectively the class and the weight associated to the training vector x_i, the discriminant function f can be expressed as:

$$f(x) = \sum_{i=1}^{M} \alpha_i y_i k(x_i, x) + b \tag{4}$$

The optimal linear separator, defined by (w^*, b^*), is chosen in order to maximize the margin ($\gamma = 1/\|w\|$) defined by the distance between the hyperplane and support vectors (x_i, $i = 1...M$ in equation 4). The optimal parameters w^* (or its corresponding vector α^*) and b^* represent the solution of the primal optimization problem:

$$\begin{aligned} \min_{w,b} \quad & \langle w, w \rangle & (5) \\ \text{subject to} \quad & y_i \left(\langle w, \phi\left(x_i\right) \rangle + b \right) \geq 1, \ i = 1, ..., n \end{aligned}$$

Transforming this optimization problem to its dual form, the optimal squared inverse margin $\omega(K) = 1/\gamma^2$ corresponding to the Gram matrix K can be expressed as follows:

$$\begin{aligned} \omega(K) \ &= \ \langle w^*, w^* \rangle & (6) \\ &= \max_{\alpha} \left(2\alpha^t \mathbf{1} - \alpha^t G(K) \alpha \right) & (7) \\ & \quad \text{subject to} \ \ \alpha \geq 0, \ \alpha^t y = 0 \end{aligned}$$

Here $\mathbf{1}$ is the n dimensional vector of ones, $\alpha \in \mathbb{R}^n$, K is the $n \times n$ Gram matrix of the n training vectors ($K_{ij} = k(x_i, x_j)$), $G(K)$ is defined by $G_{ij}(K) = K_{ij} y_i y_j$ and $\alpha \geq 0$ means $\alpha_i \geq 0, i = 1, ..., n$.

In the case of non linearly separable data, a set of slack variables is used to allow the margin constraints to be violated. The primal optimization problem (5) becomes:

$$\begin{aligned} \min_{w,b} \quad & \langle w, w \rangle + \ C \sum_{i=1}^{n} \xi_i^2 & (8) \\ \text{subject to} \quad & y_i \left(\langle w, \phi\left(x_i\right) \rangle + b \right) \geq 1 - \xi_i, \ i = 1, ..., n \\ & \xi_i \geq 0, \ i = 1, ..., n \end{aligned}$$

again by considering dual problem, the optimal solution of this problem can be expressed as:

$$\begin{aligned} \omega_C(K) \ &= \ \langle w^*, w^* \rangle + \ C \sum_{i=1}^{n} \xi_i^{2*} & (9) \\ &= \max_{\alpha} \left(2\alpha^t \mathbf{1} - \alpha^t G(K) \alpha - \frac{1}{C} \alpha^t \alpha \right) & (10) \\ & \quad \text{subject to} \ \ \alpha \geq 0, \ \alpha^t y = 0 \end{aligned}$$

where C represents a penalty parameters. We note that when $C \to +\infty$ this optimization problem is equivalent to the first one (equation 7).

4. Combining Kernel Matrix

The most important step in SVM classification systems is to define the appropriate kernel function. This function is necessary to build the Gram matrix used during training (equation 9) and testing (equation 4) steps. Many kernel functions were used for speaker verification tasks. As in score fusion approach, it will be better to combine all SVM kernel functions. We propose to build a new kernel function k_f which capitalize the information brought by each kernel. The most straightforward solution to this problem is to use a linear combination of the M base kernels:

$$k_f(x_i, x_j) = \sum_{l=1}^{M} \lambda_l k_l(x_i, y_j) \tag{11}$$

This problem has been addressed in [8] and consists in finding the parameter λ_i that maximize the optimal margin (minimize $\omega_C(K)$) over the convex cone $\mathcal{K}$ of symmetric, positive definite matrices $\mathcal{K} = \left\{ X \in \mathbb{R}^{n \times n} | X = X^t, X \succeq 0 \right\}$:

$$\begin{aligned} \min_{K \in \mathcal{K}} \max_{\alpha \in \mathbb{R}^n} \quad & \left(2\alpha^t \mathbf{1} - \alpha^t G(K) \alpha - \frac{1}{C} \alpha^t \alpha \right) & (12) \\ \text{subject to:} \quad & trace(K) = c \\ & K = \sum_{l=1}^{M} \lambda_l K_l \end{aligned}$$

where K_l is the Gram matrix corresponding to the l^{th} kernel function, M is the number of base kernels, $c \geq 0$ fixes the trace of the resulting Gram matrix. The interest of this program is that it involves in the same optimization problem the discriminant boundary (α_i, $i = 1..n$) and the weight parameters (λ_l, $l = 1, ..., M$). The output is a set of weights (λ_l, $l = 1..M$) and a discriminant function that combine information from multiple kernel space.

If we pick ($\lambda_l \geq 0$, $i = l..M$), we can omit the condition $K \in \mathcal{K}$, because this is derived from $K = \sum_{l=1}^{M} \lambda_l K_l$. The optimization problem (12) can be expressed as:

$$\begin{aligned} \min_{\lambda \in \mathbb{R}^{+M}} \max_{\alpha \in \mathbb{R}^n} \quad & \left(2\alpha^t \mathbf{1} - \sum_{l=1}^{M} \lambda_l \alpha^t G(K_l) \alpha - \frac{1}{C} \alpha^t \alpha \right) & (13) \\ \text{subject to:} \quad & \sum_{l=1}^{M} trace(K_l) \lambda_l = c \end{aligned}$$

This problem can be transposed into the following quadratically constrained quadratic program [8], whose primal-dual solution indicates the optimal weights (λ_l, $l = 1...M$) and the discriminant function (α_i, $i = 1..n$):

$$\begin{aligned} \max_{\alpha,\rho} \quad & 2\alpha^t\mathbf{1} - \frac{1}{C}\alpha^t\alpha - c\rho \qquad (14) \\ \text{subject to:} \quad & \rho \geq \frac{1}{trace(K_l)}\alpha^t G(K_l)\alpha,\ l = 1, ..., M \\ & \alpha^t y = 0 \\ & \alpha \geq 0 \end{aligned}$$

where the optimal weight λ_l corresponds to the dual variable corresponding to the l^{th} constraint in the optimization problem. This problem can be solved efficiently with programs such as SeDuMi [9] or Mosek [10].

5. SVM for Speaker Verification

An important problem of applying SVM approaches to speaker verification task is related to the variable length sequence of input speech data. They need to define a mapping of this input data to a fixed dimension vector. Different mappings have been proposed and we can distinguish two categories of methods:

The first group consists in applying SVMs using acoustic data or a mapping of it. The approach implemented in [11] trains SVMs directly on the acoustics vectors which characterize the client data and the impostors data. During testing, the segment score is obtained by averaging the scores of the SVM output for each frame. The Generalized Linear Discriminant Sequence (GLDS) kernel [12, 13] is based on an explicit mapping of each sequence to a single vector in a feature space using polynomial expansions.

The second class represents methods which use GMM adaptation methods. The MAP adaptation can be seen as a mapping of the variable length sequence of acoustic features onto a fixed vector length. All Gaussian means vectors are pooled together to get one GMM supervector. These GMM-SVM kernel functions are derived from Kullback-Leibler distance. It was proposed first in [14], and was applied for speaker verification in [15, 16] to find a separator between the speaker models and impostor models.

In our fusion experiment, we have used three different kernel functions, the first one corresponds to the GLDS kernel proposed by Campbell [12]. The last ones are the linear and non linear GMM-SVM kernels[15, 16].

5.1. Generalized linear discriminant sequence kernels

This kernel function was proposed in [12]. Given a sequence of cepstral features $x^l = (x_1, x_2, ...x_l)$, the mapping function ϕ_{glds} is expressed as:

$$\phi_{glds} : x^l \longrightarrow \frac{1}{l}\sum_{i=1}^{l} b(x_i) \qquad (15)$$

Here $b(x_i)$ is the vector of polynomial basis terms of feature vector x_i, e.g., for two features $x_i = [x_{i1}\ x_{i2}]^t$ and second order, the vector is given by:

$$b(x_i) = \left[1\ x_{i1}\ x_{i2}\ x_{i1}^2\ x_{i1}x_{i2}\ x_{i2}^2\right]^t \qquad (16)$$

The GLDS kernel function k_{glds} is defined by:

$$k_{glds}(s_a, s_b) = \phi_{glds}(s_a)^t\, R^{-1}\, \phi_{glds}(s_b) \qquad (17)$$

where $R = M^t M$ and M is defined as :

$$M = \begin{bmatrix} b(xs_1)^t \\ b(xs_2)^t \\ ... \\ b(xs_{Nspk}) \\ b(xz_1)^t \\ b(xz_2)^t \\ ... \\ b(xz_{Nimp})^t \end{bmatrix} \qquad (18)$$

where $b(xs_i)$ and $b(xz_i)$ represent respectively the expansion of speaker and impostor data (see [12] for more details).

5.2. GMM-SVM Linear kernel

The linear kernel was proposed by Campbell *et. al.* [15]. The authors used an upper bound D of Kullback-Leiber distance [17, 18] between two GMMs to build the corresponding inner product which is the kernel function as follows:

$$\begin{aligned} \mathcal{D}^2(s_a, s_b) &= \sum_{i=1}^{M} w_i \left(\mu_i^a - \mu_i^b\right)\Sigma_i^{-1}\left(\mu_i^a - \mu_i^b\right)^t \quad (19) \\ K_{lin}(s_a, s_b) &= \sum_{i=1}^{M}\left(\sqrt{w_i}\Sigma_i^{-\frac{1}{2}}\mu_i^a\right)\left(\sqrt{w_i}\Sigma_i^{-\frac{1}{2}}\mu_i^b\right)^t \quad (20) \end{aligned}$$

where w_i, μ_i^s and Σ_i are the weight, mean and covariance of each Gaussian in the s speaker GMM model.

5.3. GMM-SVM Non Linear kernel

The non linear kernel is a Gaussian kernel defined on the GMMs supervector space. It was proposed by Dehak and Chollet in [16]. The kernel function is expressed as an exponential function of distance $\mathcal{D}$ (equation 19):

$$K_{nonlin}(s_a, s_b) = e^{-\mathcal{D}^2(s_a, s_b)} \qquad (21)$$

5.4. Nuisance Attribute Projection

Nuisance Attribute Projection (NAP) [19] is a method for improving performance of SVM speaker recognition systems. The principal interest is to reduce the impact of channel, handset, session, language, etc. variations on system performances. It uses an appropriate low corank projection matrix P in the feature space to remove subspaces that cause variability in kernel:

$$K_{NAP}(x_i, x_j) = \langle P\phi(x_i), P\phi(x_j)\rangle \qquad (22)$$

6. Combining Kernel In Speaker Verification

The weight vector of linear kernel combination is computed during target speaker models training. For each base kernel function, the Gram matrix was computed using the same impostors list. The solution of the optimization problem (equation 14) provides, for each target speaker s, an optimal weight vector (λ_l^s, $l = 1..M$) and the SVM discriminant function parameters (α, b). We have picked $\lambda_i^s \geq 0$, $l = 1..M$ to avoid the test $K \succeq 0$ (K is positive definite). In the other case, we need to use test data to obtain the optimal weight that keep the kernel matrix positive definite (refer to [8] for more details). This procedure is not conform with NIST protocol, so we didn't explore yet this solution, and we were limited to the case of $\lambda_l^s \geq 0$, $i = 1..l$.

This operation is different from score fusion methods: First, we have a different weight vector for each target speaker model rather than the unique score weights vector for score fusion. The most important advantages here compared to score fusion is that we don't need extra dataset to compute the weight vector, it was computed using only training dataset(client and impostors data). In our kernel combination implementation, all kernel matrices are centered and normalized as follows [7]:

$$\text{Centering}: K_{ij} \leftarrow K_{ij} + \frac{1}{n^2}\sum_{m,o=1}^{n} K_{mo} - \frac{1}{n}\sum_{m=1}^{n}(K_{im} + K_{jm}) \quad (23)$$

$$\text{Normalization}: K_{ij} \leftarrow \frac{K_{ij}}{\sqrt{K_{ii}K_{jj}}} \quad (24)$$

This kernel combination can be seen as an adaptation of the kernel function to the speaker data. During the training task, we change the kernel function (by selecting the weight vector λ) for each speaker and find the optimal one which gives the maximal margin. The same method can be used in the case of features selection when the Gram matrices were computed on heterogeneous data. We can combine, with acoustic SVM systems, others SVM systems based for example on high-level characteristics [20] such as word usage, pronunciation, prosody, etc.

7. Experiments

7.1. Development and test databases

We performed our test experiments on the core condition of NIST 2006[2] SRE corpus (all trials). The train and test utterances contain 2.5 minutes of speech on average. The whole speaker detection task consists of 53966 tests (3612 target tests). We use equal error rate (EER) and the minimum decision cost value (minDCF) as metrics for performance evaluation.

We have used NIST 2005 [3] SRE corpus (all trials) database to train score fusion method and for tuning systems parameters. The NAP was training using a corpus extracted from NIST-SRE 2004 database.

7.2. Cepstral features

We extracted 16-dimensional Linear Frequency Cepstral Coefficients (LFCC) from speech signal every 10ms using a 20ms Hamming window. First order deltas and delta-energy are appended to the cepstral vector. Cepstral mean subtraction and variance normalization were then applied to each feature of the 33-dimensional final vector.

7.3. SVM systems

We used three SVM kernel functions in our combination. The first one is the GLDS kernel. It was constructed using the 33-dimensional vector with a 3rd degree polynomial. As in [12], the R matrix (equation 17) was approximated by using only diagonal elements to reduce the training time. The two last GMM-SVM systems used in our combination are the optimal linear and non-linear kernel obtained in [21]. In these two last systems, we have used Nuisance Attribute Projection to reduce the impact of channel and handset variations on system performances.

All SVM systems used single positive example and the same training impostors. A corpus of 449 male and 486 female impostors extracted from NIST-SRE 2004 and Fisher databases are used to train the SVM systems.

7.4. Combining kernels

We used CVX Matlab toolbox[4] (Matlab Software for Disciplined Convex Programming) with SeDuMi to train kernel combination(solve optimization problem 14). This code simultaneously solve the primal problem and its dual form. It thus returns optimal values for primal variables (α) and dual variables necessary to obtain the weight vector λ.

We performed different experiments: First, to test the influence of the parameter c (The trace of the fused Gram Matrix) on the performance of combining kernel method, we run different tests on NIST-SRE 2005 with different values of c.

Second, to compare our results with linear score fusion approaches, we have performed two different fusions: The first one consists on naive Bayes fusion approach, all subsystems scores had equal weight ($w_i = \frac{1}{M}$, $i = 1..M$). This fusion strategy was used since it has proved robust and does not require a cross validation training set. The second one is an optimal linear score fusion. In this case, the weight vector is optimized using a logistic regression (using Brummer's FOCAL toolkits[5]) [3, 4] on all 1conv4w-1conv4w trials of the NIST 2005 SRE. The resulting weight vector was used to fuse SVM systems scores on test database (NIST-SRE 2006).

8. Results and Discussion

We start by giving the results obtained for the three SVM subsystems using the three kernels (GLDS kernel, linear and Gaussian GMM supervector kernels). The Table 1 gives the EER and MinDCF of these subsystems for NIST 2005 and 2006 SRE core condition. The results show that both GMM supervector kernels perform better than GLDS kernel. These results can be explained by the fact that we apply channel compensation algorithm (NAP) only for the GMM supervector kernel systems.

Table 1: *The original subsystems performance. NIST 2005 and 2006 SRE core condition (all trials).*

	NIST 2005		NIST 2006	
System	EER	MinDcf	EER	MinDcf
GLDS kernel	9.68%	0.036	9.77%	0.045
Linear kernel	**7.38**%	0.024	6.75%	0.032
Non linear kernel	7.43%	**0.023**	**6.39%**	**0.030**

The influence of the parameter c ($trace(K)$ equation 12) on kernel combination system performances is presented in Table 2 and DET curves are plotted on figure 1. We remark first that the combining kernel method has better performances than all base kernel SVM systems for all tested values of c. So, the combining kernel takes advantage advantage of the difference between the three kernels for optimal performance. The performances of this method vary slightly (See figure 1) depending on the value of c and the best results are obtained when $c = 2$.

[2] See http://www.nist.gov/speech/tests/spk/2006/ for more details

[3] See http://www.nist.gov/speech/tests/spk/2005/ for more details

[4] http://www.stanford.edu/~boyd/cvx/

[5] See http://www.dsp.sun.ac.za/~nbrummer/focal

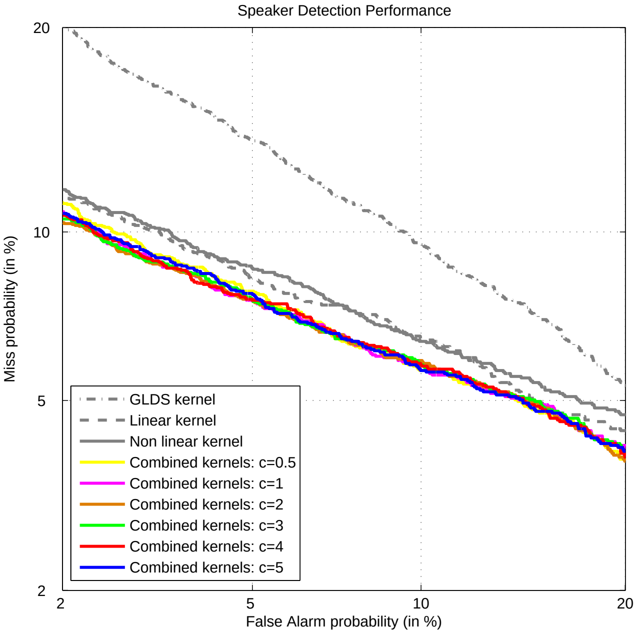

Figure 1: *DET curve of combined kernel system for different values of c, NIST 2005 core condition (all trials)*

Table 2: *The influence of the c parameter on kernel combination system performances. NIST 2005 and 2005 SRE core condition (all trials).*

	NIST 2005		NIST 2006	
System	EER	MinDcf	EER	MinDcf
0.5	6.82%	0.0226	6.24%	0.030
1	6.82%	0.0222	6.17%	0.031
2	**6.66%**	**0.0221**	**5.90%**	**0.030**
3	6.67%	0.0223	5.92%	0.031
4	6.80%	0.0222	5.98%	0.030
5	6.75%	0.0223	6.09%	0.031

There is no way to fix this parameter in advance. We use this optimal value for next comparison.

We plot on Figure 2 the DET-curves of all SVM systems and fusion systems tested on all 1conv4w-1conv4w trials of NIST-SRE 2006. The kernel combination system DET-curve are better than the three SVM systems and score fusion systems.

In Table 3, we present the performances of fusion systems. As expected, the performances of naive Bayes linear score fusion system are less than the optimal linear score fusion. The optimal score fusion performs well because the NIST SRE 2005 and NIST SRE 2006 databases are extracted from the same corpus, so there is no mismatch in data collection conditions. We obtain a little improvement (0.30% absolute) in EER. This performances are explained by the fact that all our systems used the same feature parameters with different kernel functions, we can obtain more improvements with different features.

The EER obtained using the kernel combination system is a little bit better than the naive and optimal linear score fusion systems. For MinDCF performance's, the kernel combination system is better than the naive fusion and equal to the optimal linear score fusion. These performances are explained by the fact that the kernel combination system uses a statistical criteria of maximal margin in the SVM modeling and had no prior information about the DCF function. This improvement could be more important with more SVM systems and more features.

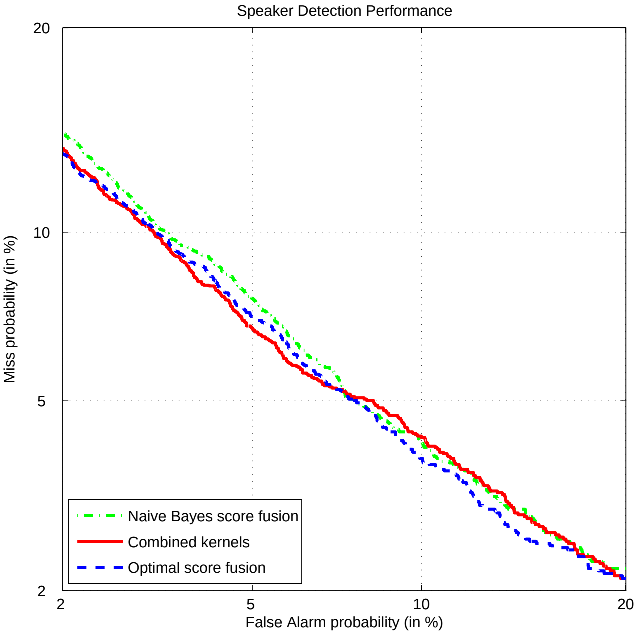

Figure 2: *DET curve of combined kernel system, naive Bayes and optimal linear score fusion. NIST 2006 core condition (all trials)*

Table 3: *Comparison between linear score fusion and linear kernel combination. NIST 2006 SRE core condition (all trials).*

System	EER	MinDcf
Naive Bayes score fusion	6.28%	0.031
Optimal linear score fusion	6.09%	0.030
Combined kernels	**5.93%**	**0.030**

9. Conclusions

In this paper, we present a new method to combine SVM speaker verification systems. This method performs a fusion in kernel function space to obtain a new SVM kernel system and we don't require extra dataset to learn the combination weights. This is an interesting advantage especially when there is a mismatch between fusion training dataset and test data. We had better performance in EER with this new method ($\sim$ 0.50% absolute improvement) than the optimal linear score fusion ($\sim$ 0.30% absolute improvement) which need a development data to estimate the fusion weight parameters. Best result was obtained with only three different kernel functions computed on the same cepstral features. We can used any other kernel function in the combination step without any modification, we need only the Gram matrices to compute the new kernel function (new Gram matrix). As it was proved with score fusion method, the performance of this approach will be even better with more SVM systems and with different features.

10. References

[1] W.M. Campbell, D.A. Reynolds, and J.P. Campbell, “Fusing Discriminative and Generative Methods for Speaker Recognition: Experiments on Switchboard and NFI/TNO Field Data,” in *Speaker Odyssey*, Toledo, Spain, June 2004, pp. 41–44.

[2] A. Stolcke, E. Shriberg, L. Ferrer, S. Kajarekar, K. Sonmez, and G. Tur, “Speech Recognition as Feature Extraction for Speaker Recognition,” in *Workshop on Signal Processing Applications for Public Security and Forensics*, Washington, D.C., 2007, pp. 39–43.

[3] P. Matejka, L. Burget, P. Schwarz, O. Glembek, M. Karaflat, F. Grezl, J. Cernocky, D.A. van leeuwen, N. Brummer, and A. Strasheim, “STBU System for the NIST 2006 Speaker Recognition Evaluation,” in *ICASSP*, Hawaii, USA, 2007.

[4] N. Brummer, L. Burget, J.H. Cernocky, O. Glembek, F. Grezl, M. Karaat, D.A. Van Leeuwen, P. Matejka, P. Schwarz, and A. Strasheim, “Fusion of Heterogeneous Speaker Recognition Systems in the STBU Submission for the NIST Speaker Recognition Evaluation 2006,” *IEEE Trans. On Audio, Speech and Language Processing*, vol. 15, no. 7, pp. 2072–2084, september 2007.

[5] W.M. Campbell, D.E. Sturim, W. Shen, D.A. Reynolds, and J. Navratil, “The MIT-LL/IBM 2006 Speaker Recognition System: High-Performance Reduced-Complexity Recognition,” in *ICASSP*, Hawaii, USA, 2007.

[6] S.S. Kajarekar, “Four Weightings and a Fusion: A Cepstral-SVM System for Speaker Recognition,” in *Proc. IEEE Speech Recognition and Understanding Workshop*, San Juan, Puerto Rico, 2005, pp. 17–22.

[7] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambrige, 2004.

[8] G.R.G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M.I. Jordan, “Learning the Kernel Matrix with Semidefinite Programming,” *Journal of Machine Learning Reasearch*, vol. 5, pp. 27–72, 2004.

[9] J.F. Sturm, “Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones,” *Optimization Methods and Software*, vol. 11–12, pp. 625–653, 1999, Special issue on Interior Point Methods (CD supplement with software).

[10] E.D. Andersen and A.D. Andersen, “The MOSEK Interior Point Optimizer for Linear Programming: An Implementation of the Homogeneous Algorithm,” in *High Performance Optimization*, H. Frenk, K. Roos, T. Terlaky, and S. Zhang, Eds. 2000, pp. 197–232, Kluwer Academic Publishers.

[11] M. Schmidt and H. Gish, “Speaker Identification via Support Vector Machines,” in *IEEE-ICASSP*, 1996, pp. 105–108.

[12] W.M. Campbell, “Generalized Linear Discriminant Sequence Kernels for Speaker Recognition,” in *IEEE-ICASSP*, 2002, vol. 1, pp. 161–164.

[13] J. Louradour and K. Daoudi, “SVM Speaker Verification using an Incomplete Cholesky Decomposition Sequence Kernel,” in *Odyssey*, 2006.

[14] P.J. Moreno, P.P. Ho, and N. Vasconcelos, “A Generative Model Based Kernel for SVM Classification in Multimedia Applications,” in *NIPS*, 2003.

[15] W.M. Campbell, D.E. Sturim, D.A. Reynolds, and A. Solomonoff, “SVM Based Speaker Verification using a GMM Supervector Kernel and NAP Variability Compensation,” in *ICASSP*, 2006, vol. 1, pp. 97–100.

[16] N. Dehak and G. Chollet, “Support Vector GMMs for Speaker Verification,” in *IEEE Odyssey*, San Juan, Puerto Rico, 2006.

[17] M.N. Do, “Fast Approximation of Kullback-Leibler Distance for Dependence Trees and Hidden Markov Models,” *IEEE Signal Processing Letters*, pp. 115–118, 2003.

[18] M. Ben and F. Bimbot, “D-MAP: a Distance-Normalized MAP Estimation of Speaker Models for Automatic Speaker Verification,” in *ICASSP*, 2003, vol. 2, pp. 69–72.

[19] A. Solomonoff, W.M. Campbell, and I. Boardman, “Advances in Channel Compensation For SVM Speaker Recognition,” in *ICASSP*, 2005, vol. 1, pp. 629–632.

[20] W.M. Campbell, J.P. Campbell, T.P. Gleason, D.A. Reynolds, and W. Shen, “Speaker verification using support vector machines and high-level features,” *IEEE Trans. on Audio, Speech and Language Processing*, vol. 15, no. 7, pp. 2085–2094, September 2007.

[21] R. Dehak, N. Dehak, P. Kenny, and P. Dumouchel, “Linear and Non Linear Kernel GMM SuperVector Machines for Speaker Verification,” in *Interspeech*, Antwerp, Belgium, 2007.